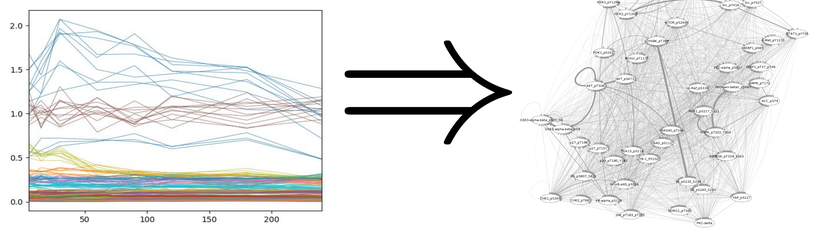
Inferring Signaling Pathways with Probabilistic Programming

David Merrell, Anthony Gitter | University of Wisconsin - Madison | Morgridge Institute for Research

1. Motivation. Signaling pathway inference

Cells regulate themselves via complicated biochemical processes called **signaling pathways.**
You can think of a signaling pathway as a **directed graph** where the nodes are proteins and the edges are regulatory relationships.
We want to infer signaling pathways from **time series phosphorylation data.**

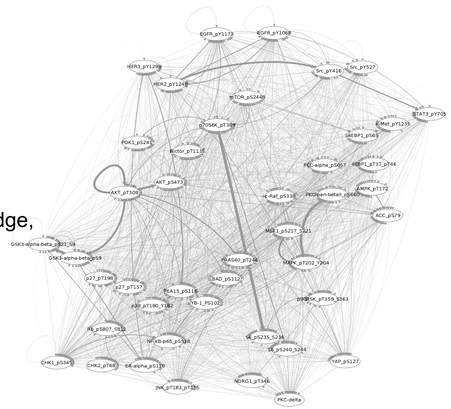


2. Formulation. DBN structure estimation

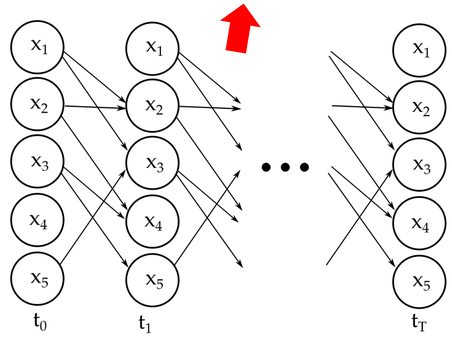
We formulate the problem as a **Dynamic Bayesian Network (DBN)** structure estimation problem.
We take a **Bayesian** approach: we want a **posterior distribution** over possible structures.

$$\underbrace{P(G|X)}_{\text{posterior}} \propto \underbrace{P(X|G)}_{\text{likelihood}} \underbrace{P(G)}_{\text{prior}}$$

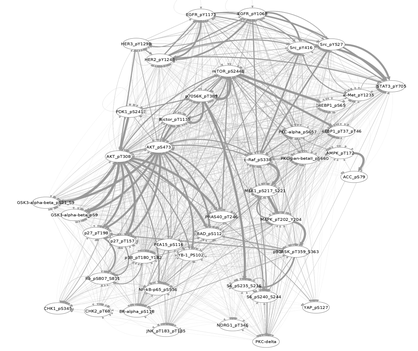
Posterior distribution. Multiplying the prior and likelihood yields an unnormalized distribution over graph structures. It updates our prior knowledge, accounting for the time series data X.



Graph likelihood function. We assume the time series data, X, are generated by a **Dynamic Bayesian Network** with structure defined by G. We marginalize the network parameters, yielding a Bayesian score for the network structure.

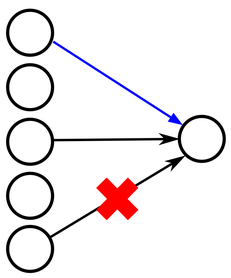


Graph Prior. We typically have prior knowledge: some relationships are more probable than others. We use this to define a prior distribution over directed graphs, G.



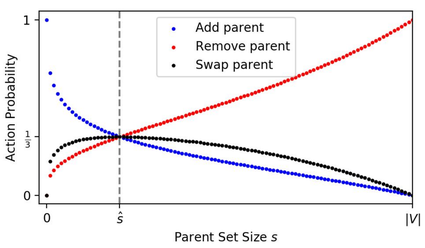
3. Algorithm. SSPS: Sparse Signaling Pathway Sampling

We call our method **SSPS.**
SSPS uses **MCMC** to sample from the posterior distribution.



We invented a **parent set proposal distribution** that efficiently explores sparse networks. It uses three actions:

1. Add-parent;
2. Remove-parent;
3. Swap-parent.



When a node has too few parents, Add-parent is most probable.
When it has too many parents, Remove-parent is most probable.
When it's "about right", Swap-parent has high probability.

4. Implementation. Probabilistic Programming

We implemented SSPS as a **probabilistic program.** Probabilistic programming is a methodology for building modular, reusable, extensible statistical models.

We wrote it in **Julia**, using the **Gen** probabilistic programming language.

Find all of our code on Github:
https://github.com/gitter-lab/ssps

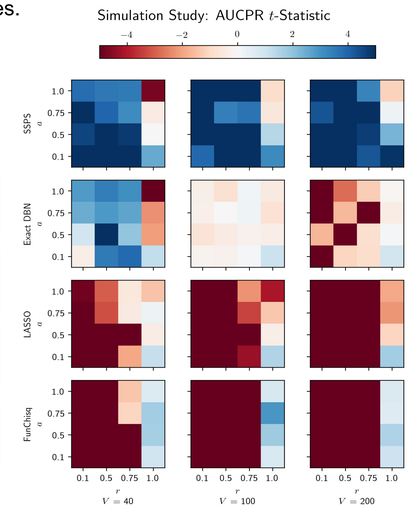
julia

Gen.jl

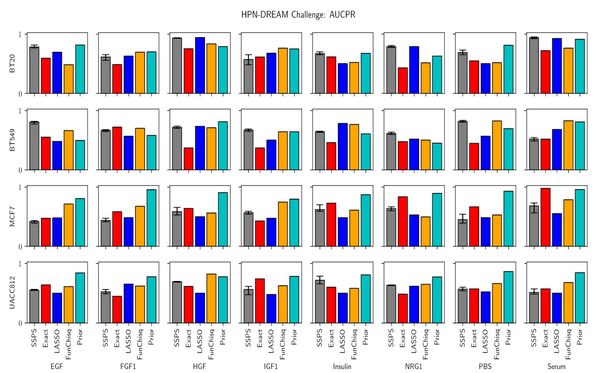
5. Evaluation. Does it actually work?

We evaluated SSPS on real and simulated data.

Simulation study. SSPS did an excellent job recovering true edges, even when the prior was corrupted. It was also much faster than similar approaches.



HPN-DREAM Challenge (Hill et al. 2016). SSPS performed about as well as existing methods. It's difficult to say anything stronger, given weaknesses in the challenge evaluation metric.



This work was funded by the **National Institutes of Health** (award T32LM012413) and the **National Science Foundation** (award DBI 1553206)